# 用 WordBasic 编程的点滴经验

何　欣　（广州金融电子化公司开发部　510110）
黄向华　（广东省计算中心开发部　510630）

**摘要**：本文介绍了在模板的设计开发过程中，如何运用 WordBasic 进行宏编程的几点经验。
**关键词**：宏　模板　控件面板　动态对话框

## 一、宏间参数的传递

若要一个宏的变量值影响另一个宏的面板控件，采用以往传统的编程方法，即在模块（函数）与模块（函数）之间传递参数变量或使用全局变量等方法是不行的。这是因为在 WordBasic 中，一个宏内的所有变量，包括全局变量也只在这个宏内有效。再加上所有变量的首次使用都不必声明，因此当一个宏的变量传递到另一个宏时，该变量实际上没有被传递，而是在另一个宏中重新分配一个被赋值为 0（变量为数值型）或被置为空（变量为字符型）的同名变量。为了解决这个问题，可采用创建临时文档的方法来传递参数。而此文档仅仅是作为一个临时文件，当参数传递完毕以后，即可被系统自动删除。该方法在面板控件相关联的具有上下级关系的两个或两个以上的面板之间的使用最为有效。下面给出一个在软件工程方面的例子来说明这种方法在实际中的运用。

例如在软件系统开发过程中，开发文档数量的多少是与软件系统规模的大小密切相关的。一般来说系统的规模越小，各开发阶段所需的文档就越少。反之则越多。假设现在有两个控件面板，控件面板一代表选择某个软件系统规模，控件面板二代表在该系统规模和开发阶段下可选的文档类型。每个控件面板就是一个宏。若选择的系统规模越小，则在相同的开发阶段（例子中假设为准备阶段）下，向用户提供的文档选项只能是部分可选的。也就是说，控件面板二的可选控件项是由前面的控件面板一的选择所决定的。

那么如何实现用临时文档在两个控件面板（宏）中传递参数呢？假设选择的系统规模是小系统。如图 1 所示：

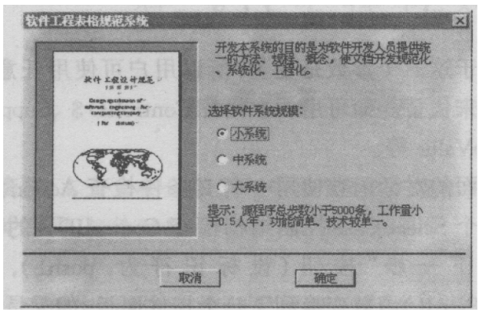


图 1　控件面板一

第一个控件面板的宏：

```
REM macro1
Sub MAIN
    .............
    Select Case xtgm //判断所选择的系统规模
      Case 0 //系统规模为小系统
        FileNew. Template = "Normal", . NewTemplate = 0//新建一个空文档
        Insert "small"//在空文档的插入点插入正文"small"(没有引号)
        Case 1 ..... //系统规模为中系统的处理
      Case 2 ..... //系统规模为大系统的处理
        Case Else
    End Select
    FileSaveAs. Name = "temp. doc", . Format = 0, . LockAnnot = 0, . Password = "", . AddToMru = 1 //保存活动文档，名为"temp. doc"
    FileClose 1 //关闭活动文档"temp. doc"
    macro2. main //激活第二个宏(第二个控件面板)
End Sub
```

假设开发阶段为准备阶段，由第一个控件面板通过临时文档 temp. doc 传递而来的参数"small"，在第二个控件面板，即第二个宏中作如下的处理：

```
Rem Macro2
Sub Main
  Dim str $ //定义字符串变量
  FileOpen. Name = "temp. doc" //打开临时文档 temp. doc
    str $ = LTrim $ (GetText $ (0, 1))//将变量 str $ 置为临时文档 temp. doc 的正文，并删除由 GetText $ ()自动加入的空格
    FileClose //关闭临时文档 temp. doc
    Kill "temp. doc" //删除临时文档 temp. doc
    If str $ = "small" Then //系统规模为小系统
      DlgEnable 1, 0 //禁止标识号为 1 的控件(即意向书选项),
    //使其仅是可见的，但不起作用
    End If
    .............
End Sub
```

执行上述的宏，控件面板一中宏的变量值就传递到了控件面板二。

处理结果如图 2 所示：

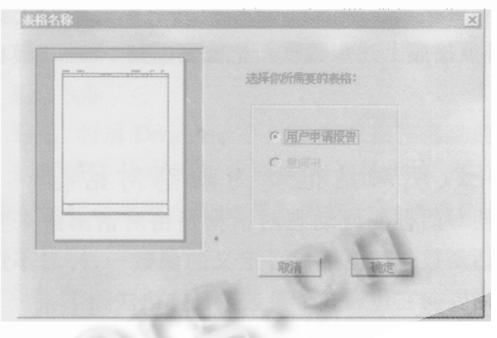


图 2　控件面板二

## 二、在不同的模板中存放宏

存放于 Normal 模板中的宏总是可用、可修改的，而存放于其他模板中的宏除非该模板已被装入，否则是不可用的。此外，存放于 Normal 模板中的宏指令越多，保存模板的时间越长。Normal 模板会因此变得太大太笨拙。若 Normal 模板过大，甚至会产生丢失 Normal 模板的现象。因此，使一个或一组宏暂时地有效是很有用的。例如：若只需偶尔地为执行的任务创建一组宏，那么这组宏就不必存放于 Normal 模板中，而只需将它们存放于另一个模板，并在需要运行它们的时候才把那组宏所在的模板用 AddAddIn 语句装入，使其作为全局模板来使用。当一个模板以这种方式被装入时，它的宏就象 Normal 模

板中的宏全局可用一样。通过使用这项技术，不仅达到了程序(宏)封装的目的，使宏在全局模板中是可读而不可见，而且较好地解决了因系统内存不足而不能运行宏的问题。

在不同的模板中存放宏的步骤如下：

1. 创建好自己的宏程序(用户创建的宏缺省存放在 Normal 模板中)；

2. 创建存放自己的宏的模板；

3. 在菜单栏上单击 File；

4. 单击“模板.....”项；

5. 在弹出的”模板和加载项”窗口中选择“管理器.....”按钮；

6. 单击“宏”书签；

7. 在“选择宏有效范围”的下拉列表中打开刚创建的模板，把创建好的在 Normal 模板中的宏复制到模板中；

8. 删除在 Normal 模板中的宏，并单击“关闭”按钮。

若要对存放在除 Normal 模板外的模板中的宏进行修改，必须按照上述步骤重新把宏复制到 Normal 模板中才可。

## 三、使对话框变为动态对话框

要使对话框变成动态对话框是由对话函数来实现的。对话函数与其他用户自定义的函数一样，只不过调用它时需要向它传递 3 个参数。具体语法如下：

```
Function FunctionName ( ControlID $ , Action, Supp-
Value)
    ..........
    FunctionName = value
End Function
```

下面是对对话函数的三个参数的解释：

ControlID $ ：接收对话框中的控件的标识串。例如：如果用户选择了对话框中的某一项，对话函数被调用，且 ControlID $ 等于该选项的标识符。

Action：标识调用对话函数的动作。共有 6 个值代表可能的动作。其中 Action 的值为 1 或 2 时较常用。当 Action 的值为 1，对应于对话框的初始化，这个值在对话框成为可见之前传递。当 Action 的值为 2，对应于用户已发出的动作，改变了对话框中控件的值(其他值的作用在此不详细描述)。

Suppvalue：接收关于一个对话框控件中变化的补充信息。

由于这三个参数是变量，所以用户可使用任意的变量名字来代替。如可用 id $ 代替 ControlID $ 、Suppval 代替 SuppValue 等。

下面的对话函数使用一个 If 条件检查 Action 的值。然后一个 Select Case 控制结构检查 ControlID $ 的值，如果是“下一步”按钮(设标识符为. push1)，调用 ChangePanel()函数变换到下一个控件面板；如果是“上一步”按钮(设标识符为. push2)，调用 ChangePanel()函数返回到上一控件面板。

```
Function f(id $ , action, suppval)
  If action = 1 Then //初始化对话框
    For j = 6 To 11
      DlgVisible j, 0 //隐藏第二个面板的控件
    Next
  ElseIf action = 2 Then //用户已作用对话框的控
                           件
    Select Case id $
      Case "push1" //如果是“下一步”按钮
        ChangePanel( nextpanel) //切换到第二个
                                  控件面板
        f = 1 //使对话框保持显示
      Case "push2" //如果是“上一步”按钮
        ChangePanel( prepanel) //返回到第一个控
                                 件面板
        f = 1
      Case Else
    End Select
  End If
End Function
```

## 四、结束语

尽管 WordBasic 是一种高级宏编程语言，函数很多，但只要掌握诀窍，宏程序设计将会相当容易、简单。另外通过对宏编程语言的使用，可使我们对宏病毒的产生有一定的了解，当我们的文档受到宏病毒的破坏时，就不会束手无策了。

(来稿时间：1998 年 3 月)